IN THE CLAIMS:

1. (Currently Amended) ~~Process~~ A method for programming an automation application program on an automation equipment programming station ~~that comprises~~, the method comprising:

[-] ~~a step to define~~ generating definitions of each of a plurality of structured ~~type~~ objects ~~(1.10, 1.20)~~ each representing an input-output channel of an input-output module of ~~the~~ an automation equipment, the definition of each structured ~~type~~ object comprising~~including~~ at least one characteristic element ~~(1.11)~~ corresponding to input-output information exchanged between the input-output module and the application program, and ~~the~~ a relative topological address of the input-output information for each said element~~,~~;

[-] ~~a step to write an application program (10),~~ declaring symbolic input-output variables ~~(100)~~ of the application program ~~(10)~~ as an instance of a previously defined structured type object~~,~~;

[-] ~~a step to configure~~ configuring symbolic input-output variables of the application program comprising a definition of ~~the~~ a physical location of each input-output module of ~~the~~ such automation equipment associated with the

symbolic input-output variables ~~(100) of the application program,~~; and

[-] ~~a step (40) to~~ automatically ~~interpret~~ interpreting the application program ~~to execute it~~for execution on ~~the~~ such automation equipment, comprising ~~a step to replace~~replacing said symbolic input-output variables ~~(100) in the application program (10) by the~~with a complete topological address ~~(200)~~ of ~~the~~ said corresponding input-output information.

2. (Currently Amended) ~~Programming process~~ The programming method according to claim 1, ~~characterized in that~~wherein each said symbolic variable ~~(100)~~ of the application program comprises two fields, a first field ~~(101) composed of~~comprising a character string chosen by ~~the~~ an application program designer, wherein the method further comprises making ~~so that~~ a structured ~~type~~ object ~~can be made to~~ correspond ~~with~~ to the symbolic variable ~~(100)~~, and a second field comprising ~~(102) composed of an~~ identification of an element of the structured ~~type~~ object associated with the symbolic variable ~~(100)~~.

3. (Currently Amended) The programming method ~~Programming process~~ according to claim 1, ~~characterized in that~~wherein said replacing said symbolic input-output variables ~~the replacement step~~ comprises:

[-] ~~a step (34) to search~~searching for the relative address defined for each structured ~~type~~ element in a table ~~(1.1, 1.2)~~ of elements of a structured ~~type~~ object stored on ~~the~~ a programming station~~,~~;

[-] ~~a step (32) to search~~searching in a configuration table for ~~the~~ said defined physical location declared for each module ~~that the designer has~~ associated with symbolic input-output variables ~~(100)~~ of the application program~~,~~; and

[-] ~~a step (33, 35) to construct the~~constructing an exact topological address of each symbolic variable ~~(100)~~ of the application program, using a means for interpretation ~~means~~ of an application program on the programming station, starting from the relative address and the physical location definition found by said searching in a configuration table.

4. (Currently Amended) The programming method ~~Programming process~~ according to claim 1, ~~characterized in that the step to define~~wherein said generating definitions of each of a plurality of structured ~~type~~ objects comprises ~~a step to create~~creating a table ~~(1.1, 1.2)~~ of structured ~~type~~ object elements comprising a first column ~~containing~~ comprising at least one identification of ~~a~~ characteristic data of the structured ~~type~~ object, a second column comprising~~containing the~~ an elementary data type (EDT) and a third column ~~containing~~ comprising the relative address of the data of the structured object, and then ~~memorizing~~ storing ~~this~~ said table in portable memory ~~means~~, for each structured ~~type~~ object.

5. (Currently Amended) The programming method ~~Programming process~~ according to claim 3, ~~characterized in that~~wherein the table ~~(1.1, 1.2)~~ of structured object ~~type~~ elements comprises a fourth column ~~containing~~ comprising a description of the data of the structured object, and a fifth column comprising~~containing~~ read or write rights for each said data of the structured object.

6. (Currently Amended) The programming method ~~Programming process~~ according to claim 1, ~~characterized in that the process comprises a step to configure~~ further comprising configuring input-output modules comprising ~~a step to select~~selecting a commercial reference of an input-output module, and ~~assignment of~~assigning an ~~the selected~~ input-output module associated with the selected commercial reference to a determined physical location, ~~the interpretation step~~said interpreting then ~~including a step to check~~comprising checking that the input-output module selected at a determined physical location is compatible with the structured ~~type~~ object configured at the same determined physical location.

7. (Currently Amended) ~~Programming~~ A programming station for programming automation equipment, comprising:

memory means ~~of memorization~~ for storing and ~~display~~displaying data~~,~~ ; and

means ~~of~~ for interaction with a designer of an automation application program ~~(10), characterized in that the programming station comprises~~;

an editor of symbolic variables ~~(100) to generate~~ for generating a configuration table ~~(6)~~ stored ~~on~~ in the memory means~~,~~ ; ~~the programming station also includes~~

~~several~~ a plurality of tables ~~(1.1, 1.2)~~ of structured ~~type~~ object elements stored ~~on~~ in the memory means~~,~~ ; and

means ~~of~~ for interpreting an application program ~~(10)~~ comprising at least one symbolic variable ~~(100)~~ defined by ~~the~~ an application designer using ~~the~~ said editor.

8. (Currently Amended) The programming station ~~Programming station~~ according to claim 7, ~~characterized in that it comprises~~further comprising means ~~of~~ for compiling ~~the~~ an application program interpreted by said interpretation means to transform the interpreted application program into an executable automation application ~~that can be executed~~for execution on an automation equipment.

9. (Currently Amended) The programming station ~~Programming station~~ according to claim 8, ~~characterized in that it comprises~~further comprising means ~~of~~ for transferring the executable automation application onto either portable memory means compatible with ~~the~~ such automation equipment, or directly onto the memory means of ~~the~~ such automation equipment.